Attorney's Docket No.: 06666/156001 / USC-3345

# APPLICATION

# FOR

# UNITED STATES LETTERS PATENT

TITLE: AUGMENTED VIRTUAL ENVIRONMENTS

APPLICANT: ULRICH NEUMANN AND SUYA YOU

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EV348187311US

September 30, 2003
Date of Deposit

# AUGMENTED VIRTUAL ENVIRONMENTS

## CROSS REFERENCE TO RELATED APPLICATIONS

**[0001]** This application claims the benefit of the priority of U.S. Provisional Application Serial No. 60/418,841, filed October 15, 2002 and entitled "AUGMENTED VIRTUAL ENVIRONMENTS (AVE) FOR VISUALIZATION AND FUSION OF DYNAMIC IMAGERY AND 3D MODELS", and is a continuation-in-part of and claims the benefit of priority of U.S. Pat. App. Serial No. 10/278,349, filed October 22, 2002 and entitled "EXTENDABLE TRACKING BY LINE AUTO-CALIBRATION".

## STATEMENT AS TO FEDERALLY SPONSORED RESEARCH

**[0002]** The invention described herein was made in the performance of work under NSF ERC EEC-9529152 SA2921 by NSF (National Science Foundation) ARO-MURI (Army Research Office - Multidisciplinary University Research Initiative), pursuant to which the Government has certain rights to the invention, and is subject to the provisions of Public Law 96-517 (35 U.S.C. 202) in which the contractor has elected to retain title.

## BACKGROUND

**[0003]** The present application describes systems and techniques relating to virtual environments, for example, augmented virtual environments.

**[0004]** Various sensing and modeling technologies offer methods for virtual environment creation and visualization. Conventional modeling systems allow programmers to generate geometric models through the manual manipulation of standard geometric primitives, libraries of pre-modeled objects, or digitizing of key points. In addition, various techniques for acquiring real world data of large environments for use in manually creating scene models have been used. Texture mapping onto the created models has also been supported, including texture mapping of static imagery onto geometric models to produce photorealistic visualizations, typically using static textures derived from fixed cameras at known or computed transformations relative to the modeled objects.

## SUMMARY

**[0005]** The present disclosure includes systems and techniques relating to augmented virtual environments. According to an aspect, a three dimensional model of an environment can be generated from range sensor information representing a height field for the environment. Position and orientation information of at least one image sensor in the environment can be tracked with respect to the three dimensional model in real-time. Real-time video imagery information from the at least one image sensor can be projected onto the three dimensional model based on the tracked position and orientation information, and the three dimensional model can be visualized with the projected real-time video imagery. Additionally, generating the three dimensional model can involve parametric fitting of geometric primitives to the range sensor information.

**[0006]** According to another aspect, a three dimensional model of an environment can be obtained. A region in motion with respect to a background image in real-time video imagery information from at least one image sensor can be identified in real time, the background image being a single distribution background dynamically modeled from a time average of the real-time video imagery information. A surface that corresponds to the moving region can be placed in the three dimensional model. The real-time video imagery information can be projected onto the three dimensional model, including the surface, based on position and orientation information of the sensor, and the three dimensional model can be visualized with the projected real-time video imagery.

**[0007]** Additionally, placing the surface can involve casting a ray from an optical center, corresponding to the real-time video imagery information, to a bottom point of the moving region in an image plane in the three dimensional model, and determining a position, an orientation and a size of the two dimensional surface based on the ray, a ground plane in the three dimensional model, and the moving region. Identifying the region in motion in real time can involve subtracting the background image from the real-time video imagery information, identifying a foreground object in the subtracted real-time video imagery information, validating the foreground object by correlation matching between identified objects in neighboring image frames, and outputting the validated foreground object.

## DRAWING DESCRIPTIONS

**[0008]** FIG. 1 is a block diagram illustrating an augmented virtual environment system.

**[0009]** FIG. 2 is a block diagram illustrating a portable data acquisition and tracking system.

**[0010]** FIG. 3 is a flowchart illustrating a process that generates an augmented virtual environment.

**[0011]** FIG. 4 is a flowchart illustrating a process of projecting real-time video imagery.

**[0012]** FIG. 5 is a flowchart illustrating a process that generates a three dimensional model of an environment.

**[0013]** FIGS. 6-9 illustrate projection of real-time video imagery information onto a three dimensional model.

**[0014]** FIG. 10 shows three sensors projecting onto one model simultaneously.

**[0015]** FIGS. 11-13 show rendered views from the three camera (projection) viewpoints of FIG. 10.

**[0016]** FIG. 14 is a block diagram illustrating another augmented virtual environment system.

**[0017]** FIG. 15 is a flowchart illustrating a process of analyzing video imagery to segment and track moving objects in a scene in real time.

**[0018]** FIG. 16 is a perspective view illustrating a process of using a segmented region of video imagery to determine and dynamically position an approximate model of a moving object in a three dimensional environment model to capture the projection of the segmented imagery.

**[0019]** Details of one or more embodiments are set forth in the accompanying drawings and the description below. Other features and advantages may be apparent from the description and drawings, and from the claims.

## DETAILED DESCRIPTION

**[0020]** FIG. 1 is a block diagram illustrating an augmented virtual environment system 100. The system 100 can include range sensors 110, image sensors 120, and tracking sensors 130. These sensors 110, 120, 130 can feed information to various components of the system 100, including a data acquisition component 140 and a motion tracking component 150.

**[0021]** The data acquisition component 140 can supply information to a model reconstruction component 160, and the motion tracking component 150 can supply information to a model refinement component 170. The model reconstruction component 160 and the model refinement component 170 can interact with each other to improve the three-dimensional (3D) model continuously during generation of the augmented virtual environment. The image sensors 120 and the model reconstruction and refinement components 160, 170 can feed a dynamic fusion imagery projection component 180, which can supply augmented virtual environment information to a visualization sub-system 190 to create the augmented virtual environment.

**[0022]** The system 100 can generate rapid and reliable realistic three-dimensional virtual environment models, which can be used in many different virtual environment applications, including in engineering, mission planning, training simulations, entertainment, tactical planning, and military operations in battlefield environments. Real-time texture images can be obtained using the image sensors 120 and mapped onto a created 3D model, providing a dynamic and up-to-date picture of the modeled environment. This can be of particular value in time-critical applications, such as military command and control, person or vehicle tracking, and catastrophe management. The system 100 can provide a rapid an accurate fusion of dynamic appearance and geometric data using real-time video and other sensor data.

**[0023]** The augmented virtual environment (AVE) system 100 can capture, represent, and visualize dynamic spatio-temporal events and changes within a real environment. The AVE system 100 can provide virtual-reality augmented by the fusion of dynamic imagery onto a 3D model using integrated systems and techniques, including model reconstruction, model refinement, building extraction, sensor tracking, real-time video/image acquisition, and dynamic texture projection for 3D visualization, as described further below. The AVE system 100 can rapidly create realistic geometric models and dynamically refine and update these models based on multiple sensor sources using the described techniques, allowing both the geometric information and the appearance of the virtual environments to be accurate and realistic analogues of the real world. Multiple streams of dynamic video imagery can be fused on the 3D model substrate to provide a coherent data visualization that enhances scene comprehension, object tracking, and event detection. This can assist users viewing imagery in the AVE system to comprehend and interpret a large number of concurrent image streams from multiple cameras moving within a scene.

**[0024]** The range sensors 110 can include one or more active airborne laser sensors to quickly collect 3D geometric samples of an environment, and initial model acquisition can be performed using the information from these laser sensor(s). Raw data samples can be processed into a consistent 3D geometric mesh model of an environment. For example, a LiDAR (light Detection and Ranging) sensor system can be used in an aircraft flyover to quickly collect a height field for a large environment. The collected data can have an accuracy of centimeters in height and sub-meter in ground position, and higher accuracies can be obtained with newer laser sensor systems. Multiple aircraft passes can be merged to ensure good coverage, and the end result can be a cloud of 3D point samples.

**[0025]** The points can be projected and resampled onto a regular grid, which can be at a user-defined resolution, to produce a height field suitable for hole-filling and tessellation. The model reconstruction phase can automatically process the 3D point cloud and output the reconstructed 3D mesh model in VRML (Virtual Reality Modeling Language) format. In the system 100, the data acquisition component 140 can collect real time geometry and imagery measurements, and the model reconstruction component 160 can obtain a single 3D surface model from the sets of acquired geometric measurements.

**[0026]** The motion tracking component 150 can provide image sensor pose and motion data for registration and data fusion. This sensor tracking can facilitate the proper functioning of the system 100 as geometry and imagery are gathered by different sensor platforms. The tracked platform positions and orientations should be known for the sensor data from the multiple image sensors 120 to be properly projected into a common scene model. This can be effectively accomplished by combining a differential GPS (Global Positioning System) receiver with an off-the-shelf orientation sensor to produce a self-contained portable tracking system coupled to a video camera.

**[0027]** The six DOF (degrees of freedom) pose of a camera can be tracked by the sensors in real-time as the system moves in an open outdoor environment. The tracking data can be encoded and stored with the camera video stream in real-time, and this data can be transmitted to other parts of the system 100 for further processing and also saved on a local storage device (e.g., a hard drive on a laptop computer). The complete tracking, camera, transmission and recording system can be housed in a backpack for mobility, as described below in connection with FIG. 2.

**[0028]** The model refinement component 170 can verify and refine the reconstructed model and extract dominant scene features. An auto-calibration method can automatically estimate the 3D parameters of point and line structures and camera pose simultaneously. Lines and points are typically dominant features of man-made structures, and these features can be used to aid in improving the accuracy and stability of the camera pose estimate obtained from the tracking system. Additional details regarding these auto-calibration systems and techniques can be found in U.S. Patent Application Serial No. 10/278,349, filed October 22, 2002 and entitled "EXTENDABLE TRACKING BY LINE AUTO-CALIBRATION", which is hereby incorporated by reference.

**[0029]** The data fusion component 180 can combine all manner of models, images, video, and data in a coherent visualization to support improved understanding, information extraction, and dynamic scene analysis. The geometric models and images can be merged to create coherent visualizations of multiple image sources. An image projection method can be adapted to enable image streams of the real environment to dynamically augment the view of a 3D model. This AVE process can be complementary to that of an augmented reality (AR) in which computer graphics dynamically augment the view of a real environment.

**[0030]** The image projection can also provide an effective mechanism for scene segmentation and error identification during the model reconstruction process. This can be useful for reducing the ambiguities frequently created by noisy data, which can occur with LiDAR systems. A database used by the system 100 can incorporate geometry and temporal imagery for objects in the environment, and thus object surfaces and features can be accurately mapped from images acquired by the varying configurations of tracked cameras, thus helping in the identification and correction of model errors.

**[0031]** The visualization sub-system 190 can provide users with immersive viewing and control over the visualization. For example, the display can be an eight by ten foot screen, video-projected by a stereo video projector (e.g., a sequential-frame stereo video-projector). A tracker (e.g., a ceiling tracker) can be used to couple the rendering viewpoint to a user's head position, providing a user with a highly immersive visualization environment.

**[0032]** FIG. 3 is a flowchart illustrating a process that generates an augmented virtual environment. A three dimensional model of an environment can be generated from range sensor information representing a height field for the environment at 300. Position and orientation information of at least one image sensor can be tracked in the environment with respect to the three dimensional model in real-time at 310. Real-time video imagery information from the image sensor(s) can be projected onto the three dimensional model based on the tracked position and orientation information at 320. The three dimensional model with the projected real-time video imagery can be visualized at 330.

**[0033]** FIG. 5 is a flowchart illustrating a process that generates a three dimensional model of an environment. Points in range sensor information can be projected and resampled onto a regular grid at a user-defined resolution to produce a height field at 500. The height field can be processed using hole-filling and tessellation to generate a triangle mesh representation of the three dimensional model at 510. The range sensor information can be LiDAR model data in the form of a cloud of 3D point samples registered to a world coordinate system (e.g., Airborne Topographics Mapper data provided by Airbornl Corporation of El Segundo, CA). The raw LiDAR data can be processed by grid resampling, hole-filling, and tessellation to reconstruct a continuous 3D surface model.

**[0034]** The tessellation operation can use Delaunay triangulation, which can better preserve the topology and connectivity information of the original data than other techniques in this context. The hole-filling operation can be performed by directly interpolating the depth values in the range image in order to preserve the geometric topology of the model. To preserve the edge information, an adaptive-weighting neighborhood interpolation can be used. The interpolation weights can be determined by an inverse function of the distance between the neighbor points and the point to be interpolated. The window size of interpolation can be adaptive to the surface-hole size. When the surface-hole size is only a few points, a small window can be used that contains the close neighbors for weighing interpolation. For the large holes, the window size can be increased to ensure sufficient points for interpolation.

**[0035]** Triangle meshes can be used as the 3D geometric representation to facilitate further processing. Triangle meshes can easily be converted to many other geometric representations. Many level-of-detail techniques can operate on triangle meshes. Photometric information can easily be added to triangle mesh data in the form of texture projections. Typical graphics hardware directly supports fast image rendering from meshes.

**[0036]** Range sensor information, such as that provided by LiDAR, offers a fast and effective way to acquire models for a large environment. In urban areas, range sensor information can provide useful approximations of buildings and structures (e.g., any large urban feature). A structure can be identified in the range sensor information at 520. The range sensor information can provide a clear footprint of a building's position and height information. This global geometric information can be used to determine a building's geo-location and isolate it from the surrounding terrain.

**[0037]** Identification of buildings in the range sensor information can be based on height. For example, applying a height threshold to the reconstructed 3D mesh data can create an approximate building mask. The mask can be applied to filter all the mesh points, and those masked points can be extracted as building points. In addition to height, area coverage can also be taken into consideration in this identification process. Moreover, the height and area variables used can be set based on information known about the region being modeled.

**[0038]** A set of geometric primitives can be selected based on a shape of the structure at 530. For example, based on a building's roof shape (e.g., flat-roof, slope-roof, sphere-roof, gable-roof, etc.), the building model can be classified into several groups (e.g., a building can be classified and modeled in sections), and for each group, appropriate geometric primitives can be defined. The set of geometric primitives can be fit to the range sensor information with respect to the identified structure at 540. The geometric primitives can include linear fitting primitives, such as cubes, wedges, cylinders, polyhedrons, and spheres, and also nonlinear fitting primitives, such as superquadratics.

**[0039]** A high-order surface primitive can be useful in modeling irregular shapes and surfaces, such as classical dome-roof buildings and a coliseum or arena. Superquadrics are a family of parametric shapes that are mathematically defined as an extension of non-linear general quadric surfaces, and have the capability of describing a wide variety of irregular shapes with a small number of parameters. Superquadrics can be used as a general form to describe all the nonlinear high-order primitives, as defined in:

(1) $$r(\eta,\omega)=\begin{bmatrix} a_1\cos^{\varepsilon_1}\eta\cos^{\varepsilon_2}\omega \\ a_2\cos^{\varepsilon_1}\eta\sin^{\varepsilon_2}\omega \\ a_3\sin^{\varepsilon_1}\eta \end{bmatrix} \quad \begin{matrix} -\pi/2\le\eta\le\pi/2 \\ -\pi\le\omega\le\pi \end{matrix}$$

where $\varepsilon_1$ and $\varepsilon_2$ are the deformation parameters that control the shape of the primitive, and the parameters $a_1$, $a_2$ and $a_3$ define the primitive size in x, y and z directions respectively. By selecting different combination of these parameters, a superquadric can model a wide variety of irregular shapes, and also many standard CG (Common Graphics) primitives as well.

**[0040]** Once defined, each building primitive can be represented as a parametric model. The parametric model can describe the primitive by a small but fixed set of variable parameters. The number of parameters specified can depend on the properties of each primitive and the knowledge assumed for the model fitting. For example, a generic plane in 3D space can be represented as

(2) $$z = ax + by + c$$

which has three parameters to be estimated. However, in the case of slope-roof fitting, the parameters may be reduced from three to two by setting either parameter *a* or *b* to zero. This is based on the observation that if a building's orientation is nearly parallel to the x or y axis of defined world-coordinates, then either parameter *a* or *b* will be close to zero for most buildings, i.e., the roof of a building usually has only one slope along the x or y direction. This constraint (referred to as the zero x/y slope constraint) for slope-roof fitting can be used, and similar constraints can also be established for other primitives.

**[0041]** Building sections can be classified into several groups, in which appropriate building primitives are defined. The selected building section, i.e., a current element of interest (EOI), can be indicated in the system as a square area roughly bounding the building section. This EOI information can be provided by a user with a few mouse clicks, or the system can automatically determine the EOI information using a heuristic that processes standard deviations and derivatives of the height data.

**[0042]** The flat-roof is a typical roof type of man-made buildings, which can be modeled using the plane-primitive group, including 3D plane, cuboids, polyhedron, and the combined primitives such as hollow-cuboids. They all share the same property that the depth surface can be described by equation (2). A 3D plane primitive can be determined by two reference points and an orientation. If the building's orientation is aligned to the global direction that is defined in the working coordinates, the specified parameters can be reduced to two, i.e., each plane can be specified by two diagonal-points.

**[0043]** With the two reference points, the system automatically estimates all corners of the building roof based on the global direction. The estimated corner points can be used for detecting the roof edges using a depth discontinuity constraint. An improved 8-neighbors connectivity algorithm can be used to detect building edges. First, the geometry connectivity information of a Delaunay reconstruction can be used to track the connected edge points. Those edges that lie along the Delaunay triangulation can be accepted as the possible edge points.

**[0044]** Second, a depth filter can be used to constrain the detected edges. The depth filter can be applied to all the possible edge points, and those points having similar depth values as that of the defined reference points can pass as correct edge points. Once the roof borders have been extracted, they can be parameterized using least-square fitting, and then the roof corners can be refined again based on the fitted roof borders.

**[0045]** Plane depth fitting can be performed on all the surface points inside the roof border. The depth discontinuity constraint can be used for surface segmentation. Surface-normal information can be used, but may not be preferable due to its sensitivity to noise. The depth discontinuity constraint generally performs well for surface segmentation. After segmenting the surface points, the plane least-square fitting can be applied to the depth values of those points, and the best fitting can be the height of the refined surface.

**[0046]** Slope is a special case of the plane with non-zero horizontal or vertical normal direction. Similar to the plane primitive, a sloped roof with rectangular edges can be extracted with two reference points using the plane fitting method. The depth fitting for sloped surfaces, however, is more complex. A 3D plane defined in equation (2) has three parameters to be estimated, where the two parameters *a*, *b*, represent two slopes in the x and y directions, and the parameter *c* is an offset. But the parameters can be reduced from three to two based on the zero x/y slope constraint.

**[0047]** In the case of a building that does not meet the condition, an orientation alignment can be performed to orient the building to the reference direction. The least-square method can also be used for parameter estimation, using all the surface points inside the detected roof borders. Many roofs of real buildings frequently have two symmetric slopes. To facilitate this structure, the two connected slope primitives can be combined to form a new primitive: roof.

**[0048]** In this case, three reference points (rather than four if the two slopes were modeled separately) can be used for parameter estimations: two on the slope edges, and one on the roof ridge. The surface points of the two symmetric slopes can be segmented using the above method. The least-square fitting can be performed on the depth values of the segmented surface points for each of the two slope primitives. The accurate roof ridge can be computed based on the intersection of the two modeled slope planes.

**[0049]** Surface fitting of a generic cylinder is a nonlinear optimization problem. However, many cylinder primitives in buildings have an axis perpendicular to the ground. Based on this constraint, the rotation parameter can be eliminated from the estimate to simplify the primitive as a vertical cylinder for circle-roofs. The roof extraction and surface segmentation can be similar to the plane case, using the depth discontinuity constraint. Two concentric circles can be defined for segmentation: the inner circle for roof border detection, and the outer circle for surface point segmentation.

**[0050]** Three parameters can be used for specifying the concentric circles: one for the circle center and two for the radius. To guarantee there are enough surface points for accurate segmentation and model fitting, the defined circles should cover all the possible surface points on the rooftop. To achieve an accurate boundary reconstruction from the typically ragged mesh data, two filters can be defined to refine the detected edges: a depth filter constraining the edge points having similar depth values as that of the defined center, and a distance filter constraining the edge points to being inside of the estimated circle.

**[0051]** The depth filter can be similar as the one applied for plane primitives, but using the circle center's depth value as a filtering threshold. The distance filtering can be a recursive procedure. The detected edge points can be fit to the circle model to obtain initial estimates of the circle center and radius. These initial estimates can be used to filter the detected edges. Any edge points whose distance to the circle center is less than a threshold can then pass the filtering. The distance deviation can be used as a filtering threshold. After the distance filtering, the refined edge points can be used recursively to estimate a new border parameter.

**[0052]** The dome-roof is a common roof type in classical buildings. A simple dome-roof can be modeled as a sphere shape, but more complicated ones may need high-order surfaces to represent them. Similar to the cylinder primitive, the surface of a sphere is also a quadric surface. To detect the roof border and surface points, two reference values can be used: one for dome-roof center and another one for roof size.

**[0053]** To guarantee enough surface points for accurate segmentation and fitting, the defined area should cover all the possible points on the roof surface. Since the section-projection of a sphere is a circle, the methods for sphere-roof detection and surface segmentation can track those used for the cylinder primitive, except for not using the depth filtering as the sphere center in 3D space may not be defined. The model fitting can be performed on all the segmented spherical surface points.

**[0054]** As in the cylinder case, the distance constraint can be recursively used to achieve an accurate model reconstruction. The sphere primitive can also be combined with other type primitives. A popular usage is the sphere-cylinder combination. Another use of sphere primitives is their use as a preprocessing step for high-order surface fitting. High-order surface fitting normally is a non-linear problem. Appropriate selection of initial estimates can be useful in obtaining a convergence to an optimal solution.

**[0055]** High-order modeling primitives can be used to facilitate modeling of complex objects and irregular building structures, and may be combined with standard CG primitives. Superquadrics are a family of parametric shapes that are mathematically defined as an extension of nonlinear generic quadric surfaces. They have the capability of describing a wide variety of irregular shapes with a small number of parameters. The superquadric can be used as a general form to describe all the nonlinear high-order primitives. As an example of applying the high-order primitives to model irregular building shapes, the following is a description of modeling the Los Angeles Arena with an ellipsoid primitive. The ellipsoid is a special case of superquadrics with the deformable parameters $\varepsilon_1 = 1$, $\varepsilon_2 = 1$.

**[0056]** Object segmentation can use a region-growing approach to segment the irregular object from its background. Given a seed-point, the algorithm automatically segments the seeded region based on a defined growing rule. For example, the surface normal and depth information can be used to supervise the growing procedure. Initial surface fitting can use an appropriate initial value with the Levenberg-Marquardt (LM) algorithm to help guarantee a converged optimal solution. A sphere primitive fitting can be used for system initialization. The system can perform high-order surface filling, once initialized, by fitting the ellipsoid primitive to the segmented surface points using the LM algorithm. In this Arena modeling example, the algorithm can use six hundred and six iterations to converge to the correct solution.

**[0057]** As the models thus generated from constructive solid geometry allow the composition of complex models from basic primitives that are represented as parametric models, this approach is very general. The type of primitive is not limited and may contain objects with curved surfaces, so the flexibility of model combinations is very high. Thus, a large range of complex buildings with irregular shapes and surfaces can be modeled by combining appropriate geometry primitives and fitting strategies.

**[0058]** These techniques can be automatically applied to all buildings in the range sensor information, either in sequence or in parallel. For example, once the buildings are detected and segmented, the predefined geometric primitives can be iteratively fitted to the data and the best fitting models can be used to represent the building structures. A person can assist in selecting the group type and thus the associated primitives, or this group type selection can be fully automated. Once the group has been selected, the system can perform the primitive fitting and assembly of buildings from multiple primitives. The system can also include editing tools that allow users to modify the models or obtain a specific representation quickly and accurately.

**[0059]** The system can extract a variety of complex building structures with irregular shapes and surfaces, and this scene segmentation and model fitting approach to building a 3D model of a large urban environment can enable rapid creation of models of large environments with minimal effort and low cost. The range sensor information can be used to generate highly accurate 3D models very quickly, despite resolution limitations and measurement noise that may be present in the range sensor information. The 3D model can be easily updated and modified. Moreover, undersampled building details and occlusions from landscaping and overhangs can also be handled without resulting in data voids.

**[0060]** The three dimensional model can be refined based on object surfaces mapped from images acquired by the image sensors at 550. The dynamic image projection can provide an effective way to detect modeling errors and reduce them. By dynamically projecting imagery and/or video of the scene onto its corresponding geometric model, viewing from arbitrary viewpoints can quickly reveal model fidelity and/or errors. For example, users can verify and refine the models from different angles to cover the complete model surface; the range sensor data and the model that is fit to the range sensor data can be superimposed for visual verification, and accurate building models can be extracted.

**[0061]** Referring again to FIG. 1, tracking sensor(s) 130 can be attached to an image sensor 120, allowing the image sensor to be moved around the scene for viewpoint planning and data fusion. An image sensor 120 can be a mobile sensor platform on an aerial vehicle, a ground vehicle or a person that includes a video camera obtaining a real-time video stream. An image sensor 120 can also be a fixed position image sensor with the ability to change its orientation. Because the orientation and position of each image sensor 120 can be known, multiple video streams can be simultaneously projected onto the scene model, thereby presenting the observer with a single coherent and evolving view of the complete scene.

**[0062]** Moreover, the images acquired by the image sensors 120 can be used to both visualize dynamic changes in the scene and refine the 3D geometry; and tracking information can be used for data registration and assembly during the model reconstruction phase.

**[0063]** FIG. 2 is a block diagram illustrating a portable data acquisition and tracking system 200. The system 200 is a hybrid tracking approach that can provide a robust and precise tracking system for use in an outdoor environment. The system 200 can integrate vision, GPS, and inertial orientation sensors to provide six DOF pose tracking of a mobile platform.

**[0064]** The system 200 can be a self-contained portable tracking package (e.g., a backpack) that includes a stereo camera 210 (e.g., a real-time stereo head, such as the MEGA-D from Videre Design of Menlo Park, CA). The system 200 can include a global navigational satellite system (GNSS) receiver 230, such as a differential GPS receiver (e.g., a Z-Sensor base/mobile from Ashtech). The system 200 can also include a three DOF inertial sensor 240 (e.g., an IS300 from Intersense), and a portable data processing system 220 (e.g., a laptop computer).

**[0065]** The stereo camera 210 can be two high resolution digital cameras using a Firewire (IEEE (Institute of Electrical and Electronics Engineers) 1394) interface to the portable data processing system 220. The dual camera configuration allows one channel (e.g., the left channel) of the acquired video stream to be used for video texture projection and vision tracking processing, while the stereo stream can be used in detailed building façade reconstruction.

**[0066]** The integrated GNSS receiver 230 and inertial sensor 240 can track the system's six DOF (position and orientation) pose. The GNSS sensor can be a GPS sensor with two working modes: a differential mode and a single mode. For example, a differential GPS mode (DGPS) can include two RTK (Real-time kinematic) units (base and remote) communicating via a spread spectrum radio to perform position calculations. The DGPS can track positions to about ten centimeter accuracy. The GNSS sensor 230 can be switched between the two modes as need.

**[0067]** The inertial sensor 240 can be attached to the stereo camera 210 to continually report the camera orientation. The sensor 240 can incorporate three orthogonal gyroscopes to sense angular rates of rotation along the three perpendicular axes. The sensor 240 can also include sensors for the gravity vector and a compass to compensate for gyro drift. The measured angular rates can be integrated to obtain the three orientation measurements (yaw, pitch and roll). This orientation tracker can achieve approximately one degree RMS static accuracy and three degrees RMS dynamic accuracy, with a 150 Hz maximum update rate.

**[0068]** The tracking system 200 can run in real-time including sensor and video acquisition, storage, and transmission. The sensors in the tracking system 200 may run at different updates rates (e.g., the DGPS may update at about one Hz, the inertial sensor may run at 150 Hz, and the video camera frame rate may be 30 Hz). This can be compensated for by synchronizing and resampling all three data streams at a common rate (e.g., the 30 Hz video rate).

**[0069]** In order to maintain accurate registration between the geometric models and the real video textures, a complementary vision tracker can be used to stabilize the tracked camera pose. The tracked camera pose defines the virtual video projector used to project texture images onto the 3D geometric models, so pose tracking accuracy can directly determine the visually perceived accuracy of projected texture registration. Moreover, vision tracking can also be used to overcome cases of GNSS dropouts caused by satellite visibility occlusions.

**[0070]** The vision tracker can be based on feature auto-calibration, adapted for this application. Vision tracking can use an Extended Kalman Filter (EKF) framework that can extend tracking range from an initial calibrated area to neighboring uncalibrated areas. Starting from a known initial estimate of camera pose, which can be obtained from any method or sensors (e.g., the GNSS-inertial tracker), the camera pose can be continually estimated using naturally occurring scene features based on a prediction-correction strategy.

**[0071]** Both line and point features of a scene can be used for tracking. Straight lines are typically prominent features in most man-made environments, and can be detected and tracked reliably. The line feature can be modeled as an infinite straight line, and its observed line segments in different views may correspond to different portions of the same line. Point features are also useful for tracking, especially when the user is close to scene surfaces, in which case, there may not be lines visible for pose estimation.

**[0072]** An EKF can estimate pose based on both line and point features. The camera state can be represented as a six dimension vector of position, incremental orientation, and their first derivatives:

$$[x, y, z, x', y', z', \Delta\theta, \Delta\vartheta, \Delta\psi, \theta', \vartheta', \psi']$$

The linear dynamic model can be used for state prediction, and both line and point features can be used as measurements for the EKF state update. For every new frame, the tracker can first predict the camera pose based on the prediction equation. The model features can then be projected back on the image plane based on the prediction, and the discrepancy between projected features and observed features can be used to refine the prediction.

**[0073]** Once calibrated, the camera internal parameters can be assumed fixed during the tracking session. Using these vision tracking techniques, observed image elements can properly retain their alignment with the 3D positions and orientations of model objects as cameras change their viewpoints.

**[0074]** Referring again to FIG. 1, the dynamic fusion imagery projection component 180 can dynamically fuse the data of multiple spatio-temporal information sources from geometric models, images, video and other sensing information. Texture images can be associated with the sensor and the sensor's pose within the model. The mapping between the model and the image can be created dynamically as a result of image projection during the rendering process. Thus, changing the model and sensor relationships automatically changes the mapping function and associated imagery, and also the relationships of visibility and occlusion. As new images arrive, or as the scene changes, the sensor pose and image information can simply be updated, rather than repeating the time consuming process of finding mapping transformations.

**[0075]** The system 100 can use a model of perspective image projection, a strategy to handle the problem of visibility and occlusion detection, and an accurate sensor model including camera parameters, projection geometry, and pose information. Projective texture mapping can be used by the system 100. This technique models the process of perspective image projection by casting the lines of light onto objects using the light source position as the projection center.

**[0076]** The texture coordinates can thus be automatically assigned to model surface points in the rendering process. By specifying an image of the scene as a texture and specifying the camera pose, the image can be projected back onto the scene geometrically. As the camera moves, its changing pose and imagery can dynamically paint the acquired images onto the model; the projector parameters are the same as the camera parameters: viewing transform and view volume clipping. These allow for dynamic control during rendering.

**[0077]** FIGS. 6-9 illustrate projection of real-time video imagery information onto a three dimensional model. FIG. 6 illustrates a modeled building structure 600 and a projection frustum 610 created by a known camera pose. FIG. 7 illustrates the initial camera image 700. FIG. 8 illustrates projection 800 of the image onto the model. FIG. 9 illustrates the rendered view 900 in an AVE system.

**[0078]** While image projection is a powerful approach to integrating dynamic imagery with 3D models, the issue of occlusions should be taken into consideration. Referring again to FIG. 1, the projection component 180 can ensure that the projections texture the surfaces visible to the camera that captured the images. Thus, the projection component 180 can modulate the projection process with visibility information.

**[0079]** Visibility detection should be fast in order to achieve a real-time visualization session. Depth-map shadows can be used as an approach to fast visibility detection that is supported by many high performance graphics cards, such as NVIDA's Geforce 3 GPU, which supports 24-bit shadow maps. The depth-map shadows process produces a depth map that is used as a texture image. Based on a comparison of the projected depth value against the *r* component of texture coordinates, the projection component 180 can determine if the surface point is visible or hidden from an image sensor 120.

**[0080]** FIG. 4 is a flowchart illustrating a process of projecting real-time video imagery. A depth map image can be generated from a video sensor viewpoint at 400. Then real-time video imagery information can be projective texture mapped onto the three dimensional model conditioned upon visibility as determined from the depth map image at 410. This approach represents a two-pass process: a first pass for generating the depth image needed for comparisons, and a second pass for image projection, conditional upon visibility testing. However, this can be implemented as a one-pass approach utilizing graphics hardware that supports SGI OpenGL extensions.

**[0081]** Referring again to FIG. 1, a P4 2G system using this approach can achieve real time rendering at 26 Hz of 1280x1024 images with four texture streams projected onto the 3D model. By including the sensor information into the AVE database, the system 100 permits users to dynamically control the sensor within the virtual environment scene and simultaneously view the information visible to that sensor mapped onto a geometric model from an arbitrary viewpoint.

**[0082]** The projection process can involve loading the video imagery and sensor pose streams into memory buffers and converting them to the formats used for texturing and projection. The video and sensor streams can be synchronized during the data acquisition process so that each frame of the video has a timed sensor tag associated with it, including the sensor's internal parameters and 6DOF pose information. The sensors, geometric model and images can come from multiple modalities with different coordinate reference, and these can be converted constantly into a common coordinate reference that is defined in the model system.

**[0083]** Once in the memory buffers, the data is ready for processing and texturing. The sensor's internal parameters can be used for specifying the virtual projector's projection, and the pose data can define the projector's position and orientation. This process can sequentially use model and projection matrix operations. Sequentially specifying multiple projectors allows simultaneous projection of multiple images onto the same model.

**[0084]** Projection visibility can be computed using the depth shadow mapping method for each sensor from its viewpoint. The occluded portions can keep their original colors or blend with other projections, depending on application and user preferences. After texture coordinates generation and projective mapping, the projection can be clipped to the sensor's viewing frustum specified by the sensor parameters. This can be done using the stencil operations: drawing the outside of the frustum with a stencil increment operation, and the inside with a stencil decrement operation, therefore masking the projection pass to the regions of the screen where the scene falls within the sensor frustum.

**[0085]** The visualization sub-system 190 can include a virtual reality display headset, a large 3D visualization screen, and/or a quad-window display interface to maximize visualization effectiveness. The quad-window display can simultaneously show the 3D geometric model, the 2D video image prior to projection onto the 3D model, the fusion result viewed from a novel viewpoint, and a rendered view from an image sensor viewpoint. The four windows can be linked to each other and can be interactively controlled during visualization sessions. The large 3D visualization screen can be an 8x10 foot screen, video-projected by a stereo video-projector. A 3rdTech ceiling tracker can be used to couple the rendering viewpoint to user's head position. A tracker can also facilitate mouse-like interactions.

**[0086]** Additionally, the projection component 180 can fuse multiple simultaneous image streams onto the 3D model. The system 100 provides the 3D substrate model that ties the multiple streams together and enables a person to readily understand images from multiple cameras, which may be moving as well. The system 100 can support multiple texture projectors that can simultaneously visualize many images projected on the same model. This can result in a coherent presentation of the multiple image streams that allows the user of the system 100 to easily understand relationships and switch focus between levels of detail and specific spatial or temporal aspects of the data.

**[0087]** The AVE fusion presentation can present huge amounts of visual data at varying levels of detail all tied to a reference model that makes the activities within and between multiple images comprehensible.

**[0088]** FIG. 10 shows three sensors projecting onto one model simultaneously. Three video projections 1010, 1020, 1030 within one model area 1000 are illustrated, creating an augmented virtual environment that allows a person to visualize and comprehend multiple streams of temporal data and imagery. FIG. 11 shows a rendered view from the camera (projection) viewpoint 1010. FIG. 12 shows a rendered view from the camera (projection) viewpoint 1020. FIG. 13 shows a rendered view from the camera (projection) viewpoint 1030.

**[0089]** FIG. 14 is a block diagram illustrating another augmented virtual environment system 1400. The system 1400 can include components 110-170, 190 and function as the system 100 described above in connection with FIG. 1, but also includes the ability to model dynamic geometry and appearance changes of moving objects, such as people and vehicles, within the scene. The inclusion of dynamic models in an augmented virtual environment can improve complex event visualization and comprehension, such as in command and control and surveillance applications.

**[0090]** An object detection and tracking component 1410 can derive dynamic aspects of the model from real-time video imagery and other sensor data gathered from multiple, and possibly moving sources in the scene. The component 1410 can detect and track moving regions in real-time video imagery to facilitate the modeling of dynamic scene elements in real-time. The output of the object detection and tracking component 1410 can be provided, along with the real-time video imagery, to a dynamic fusion imagery projection component 1420. The fusion component 1420 can perform as the fusion component 180 described above, and the fusion component 1420 can also dynamically add moving objects in real time to the three dimensional model based on the output of the object detection and tracking component 1410.

**[0091]** The object detection and tracking component 1410 and the dynamic fusion imagery projection component 1420 can be used with other system configurations as well. For example, the three dimensional model of the environment can be generated differently, including being generated elsewhere and being obtained (e.g., loaded from disk) by the system 1400. Moreover, the image sensors need not be mobile sensors with the described position and orientation tracking capabilities, but may rather be fixed image sensors with known positions and orientations.

**[0092]** FIG. 15 is a flowchart illustrating a process of analyzing video imagery to segment and track moving objects in a scene in real time. A background image can be estimated at 1510 by modeling the background image as a temporal pixel average of real-time video imagery information (e.g., an input video sequence 1500). The background image can be subtracted from the real-time video imagery information at 1520.

**[0093]** The background estimation can largely determine the performance of the overall system. A variable-length time average can dynamically model a single distribution background, as opposed to using a Gaussian based background model.

Detecting foreground objects involves characterizing the similarity between new observations and the background model. The Euclidean distance of the measurements can be used as the similarity criteria:

$$(3) \quad \Delta I_i(x,y) = \left\| I_i(x,y) - B_k(x,y) \right\|$$

where $\Delta I_i(x,y)$ is the difference image, and $I_i(x,y)$ is the input image at frame *i*. $B_k(x,y)$ is the modeled background image, which can be dynamically updated at selected time interval *k*, which can be from several seconds to several days.

**[0094]** The background image can be modeled as a temporal pixel average of recent history frames:

$$(4) \quad B_k(x,y) = \frac{1}{N}\sum_{n=0}^{N} I_{i-n}(x,y),$$

where *N* is the number of frames used for calculating the background model. *N* can be set to five, which can offer performance similar to that of a single Gaussian distribution, with lower computation complexity.

**[0095]** One or more foreground object(s) can be identified in the subtracted real-time video imagery information using a histogram-based threshold and a morphological noise filter at 1530. In general, a pixel can be considered a foreground candidate if its difference value is above a pre-defined threshold. Otherwise the pixel can be classified as background.

**[0096]** The parameters of this foreground-background classification can be estimated by taking the quality of the video recorders and the size of the foreground target into consideration. The threshold can be pre-trained off-line, and the optimal value can be obtained such that only the top five percent differences are labeled as potential foreground objects. From the classified results, a mask image can be constructed by using a two-pass 4-neighbors connectivity algorithm. The mask image can be used to filter the input image to derive the foreground object regions.

**[0097]** To reduce noise, a morphological filter can be used to constrain the segmented objects. The filter can be applied to all the possible foreground regions, and only those regions with areas large enough are then passed as correct foreground objects. For example, objects smaller than 0.1 percent of the image size can be filtered out as noise.

**[0098]** The identified foreground object(s) can be validated by correlation matching between identified objects in neighboring image frames at 1540. This can be understood as pseudo-tracking of moving objects over time to confirm that they should be modeled separately in the environment. Objects that are validated for less than a second can be eliminated. This can eliminate spurious new object detections, such as for waiving trees or other foliage.

**[0099]** A normalized intensity correlation can be used as the matching criteria to measure the similarities between two detected regions. The sum of squared error of a 10x10 window located at the center of the objects can be evaluated, and the minimum error pairs can be selected as the best tracked matches. Object regions with an acceptable matching score can be assigned a target tag number. The correlation threshold can be determined experimentally by examples and by taking the quality of the video cameras and imaging conditions into consideration.

**[0100]** The validated foreground object(s) can be output at 1550. The final outputs of the detection and tracking system can be the four-corner coordinates $\{x_i \mid x_i = (x_i, y_i), i = 1 \sim 4\}$ bounding the moving object regions in the 2D image plane. This tracking information can be used for modeling the moving objects and estimating their relative 3D positions in the scene.

**[0101]** FIG. 16 is a perspective view illustrating a process of using a segmented region of video imagery to determine and dynamically position an approximate model of the moving object in a three dimensional environment model to capture the projection of the segmented imagery. FIG. 16 illustrates the case of a ground-level camera and a tracked object (e.g., a person or vehicle) that rests on the ground 1600 in a three dimensional model. A dynamic model surface 1660 can be placed in the three dimensional model to capture the projection of an identified moving object.

**[0102]** The surface 1660 can be a two dimensional surface or a three dimensional surface. The surface 1660 can be a complex shape that closely fits the identified moving region. For example, a complex polygon can be used, where the polygon has many little edges that follow the outline of the moving object; or a complex 3D shape can be used to more accurately model a moving object (e.g., a 3D surface defined by multiple cubes for a car, and a 3D surface defined by multiple cylinders for a person). In general a simple 2D polygon can be sufficient to create a dynamic object model in the three dimensional environment model that enhances the visualization of moving objects in the augmented virtual environment, and use of a 2D polygon can require less processing in the system.

**[0103]** In the illustrated example, three parameters can define the dynamic model polygon: position, orientation and size. A point 1610 is the 3D optical center of a camera generating an image being processed. A bounding shape 1630 can be a bounding box in an image plane 1620 of the tracked moving object. A point 1640 can be the mid-point of the bottom edge of the bounding box.

**[0104]** Using point 1610 and point 1640, a ray (or vector) 1650 can be cast to intersect with the ground 1600, and the intersection point 1670 can be identified. If no intersection point exists, or if the ray 1650 intersects with a building in the 3D model before it intersects the ground, the tracked object can be identified as an airborne object and dealt with separately using other sensor data, or simply be disregarded for purposes of moving object modeling.

**[0105]** With the assumption that moving objects rest on the ground, point 1670 can be used to define the 3D position of the model surface 1660. The orientation of the model surface 1660 is denoted by a vector 1680, which can be set in the same vertical plane as the vector 1650 and set perpendicular to a scene model up-vector (i.e., the vector 1680 lies in the ground plane 1600). Two corners of the bounding box 1630 can then be projected onto the model plane to determine the size of the model surface 1660.

**[0106]** This creation of a dynamic model for moving objects can address the distortion that occurs when projecting video containing moving objects onto a static model. A video sensor can capture the dynamic movements of a person or a car in the scene. Dynamic models of the moving objects in the scene can be created and added to an AVE visualization as described to reduce visual distortions. The video texture is projected onto the model as before. Users can freely navigate around the complete model.

**[0107]** The logic flow depicted in FIGS. 3-5 and 15 do not require the particular order shown. Many variations are possible, sequential order of operations is not required, and in certain embodiments, multi-tasking and parallel processing may be preferable.

**[0108]** Other embodiments may be within the scope of the following claims.